Optimal Statistical Tuning of the RED Parameters

Homayoun Yousefi'zadeh Amir Habibi Hamid Jafarkhani
Department of EECS
University of California, Irvine
[hyousefi,ahabibi1,hamidj]@uci.edu

Claus Bauer
Dolby Laboratories
San Francisco, CA
cb@dolby.com

***Abstract*— Achieving minimal loss while satisfying an acceptable delay profile remains to be an open problem under the RED queuing discipline. In this paper, we present a framework targeted at optimal fine tuning of the RED parameters in order to address such problem. For a given traffic pattern and utilizing a statistical analysis of finite-state Markov chains, we formulate an optimization problem aimed at addressing the loss and delay tradeoff of the RED queuing discipline. Our two-step iterative solution to the problem identifies the optimal settings of the RED parameters. We prove the convergence of our solution and investigate its low complexity characteristics. We apply our framework to a number of generic queuing and TCP scenarios in order to capture loss and delay performance of our algorithms versus buffer capacity and service rate. Based on our results, we argue that our model is capable of optimally addressing the loss-delay tradeoff of RED queues accommodating time-varying traffic profiles.**

***Index Terms*— RED, Markov Chain, Optimal Parameter Fine Tuning, Packet Loss, Queuing Delay.**

I. Introduction

In the past years, Active Queue Management (AQM) schemes [5] have been proposed as key schemes to prevent excessive loss of Internet traffic. Random Early Drop (RED) [11] and Random Early Marking (REM) [15] are arguably the most widely studied AQM schemes. While both schemes follow the same concept of operation pertaining to early detection of a congestion phenomenon, RED relies on intermediate nodes to react to a congestion phenomenon rather than the end nodes utilized by REM.

Although random early detection schemes can potentially outperform traditional drop-tail schemes, it is often difficult to parameterize random early detection queues under different congestion scenarios. In addition, there is a need for constant fine tuning of parameters to adapt to current network conditions. To that end and based on simplified models, guidelines have been proposed in [7], [6], [21] for setting RED parameters. However, most studies on RED are based on heuristics or simulations rather than a systematic approach. Of the literature articles, the authors of [2] and [9] have modeled RED stochastically, while those of [14], [20], and [19] have used a Markovian model to study RED.

In this paper, we perform a systematic study on the optimal fine tuning of RED parameters. The parameters of interest include the two thresholds, the probability of drop in the intermediate regime, and the instantaneous queue weighting function. Given the statistical properties of the arrival pattern of a RED queue, our objective is to minimize its loss characteristic while satisfying an acceptable delay profile. The formulation of our problem appears in the form of a constraint optimization problem that can be efficiently solved in two iterative steps.

This paper is structured as follows. In Section II, we discuss RED preliminaries. In Section III, we formulate and solve our optimization problem. We consider two cases associated with instantaneous and average queue sizes observed in a RED queue. Section IV provides our simulation results applied to both generic queuing and TCP scenarios. Finally, Section V concludes this paper.

II. RED Preliminaries

In this section, we describe the RED algorithm and its associated queuing models in conjunction with the arriving traffic profile of the queue. Our objective is to identify the steady-state distribution probability of the occupancy of a given buffer the behavior of which is governed by RED.

A. The RED Algorithm

The average queue size of a RED queue is calculated using a low-pass filter with an exponential weighted moving average as

$$q_t = (1 - w_q)\, q_{t-1} + w_q\, \tilde{q}_t \tag{1}$$

where q_t is the current average queue size, q_{t-1} is the average queue size at the last time instant, w_q is the weighting function, and $\tilde{q}_t$ is the current instantaneous queue size. q_t is then compared to two thresholds, a minimum threshold q_{min} and a maximum threshold q_{max}. Each arriving packet is dropped with probability p given by

$$p = \begin{cases} 0, & \text{if} \quad q_t < q_{min} \\ \epsilon_i = \frac{q - q_{min}}{q_{max} - q_{min}}\, p_{max}, & \text{if} \quad q_{min} \le q_t < q_{max} \\ 1, & \text{if} \quad q_t \ge q_{max} \end{cases} \tag{2}$$

While in our study p is varied linearly from 0 to p_{max} in the region between q_{min} and q_{max}, there are many other possibilities of choosing this drop probability. Examples include choosing p as a nonlinear convex, or nonlinear concave function of the queue size.

B. Traffic Profiling

We consider a queuing system with the capacity of K fixed size packets operating under the RED queuing discipline. We note that a fixed packet size can represent an atomic unit of operation when receiving variable length packets and thus does not represent a loss of generality. The RED queuing system is described by its traffic pattern and is assumed to be operating in its steady-state regime. In [12] and [13], the authors develop a model for analyzing both transient and steady-state behavior of

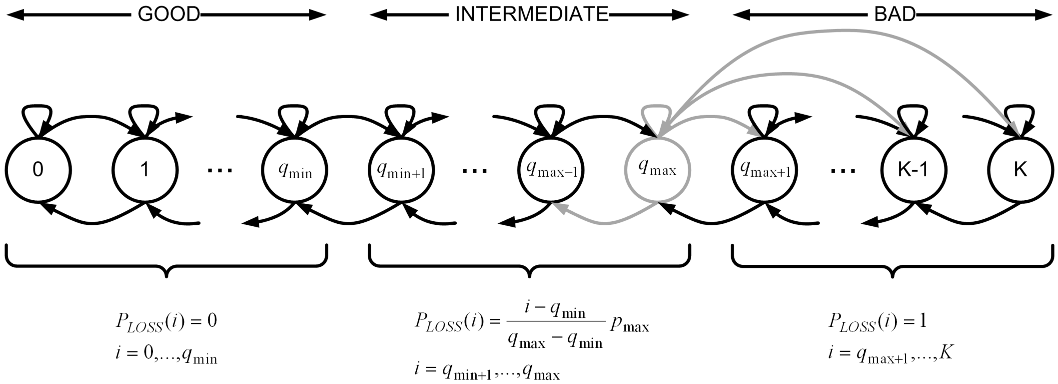


Fig. 1. M/D/1/K approximation of the steady-state behavior of RED under quasi-stationary assumptions. For clarity, only the full set of transitions associated with $k = q_{max}$ are shown.

RED queues accommodating a large population of random traffic sources the traffic generation pattern of which is described by a Poisson arrival process with a time varying rate. As the result of enforcing RED packet discarding mechanism with a small averaging factor of w_q in the order of 10^{-3} and for a slowly varying Poisson parameter, the RED queue is considered to be operating in a quasi-stationary state. As such, the behavior of the queue can be approximated with M/G/1/K queuing discipline. For the purpose of our study, we select the M/D/1/K queuing discipline not only as a special case of M/G/1/K but as the best practical alternative considering the fact that today's Internet network buffers provide a fixed deterministic service rate. Fig. 1 shows such queuing system.

For an M/D/1/K queue with a load factor ρ, we normalize the service time to indicate the time unit such that the arrival intensity is equal to ρ. Then, the steady-state probabilities π_k of being in state k for $k \in \{1, \cdots, K\}$ form a discrete Probability Density Function (PDF) the terms of which are calculated as

$$\pi_k = \begin{cases} \frac{\pi_k^\infty}{\pi_0^\infty + \rho G(K)}, & \text{if} \quad k \in \{0, \cdots, K-1\} \\ 1 - \frac{G(K)}{\pi_0^\infty + \rho G(K)}, & \text{if} \quad k = K \end{cases} \quad (3)$$

where $G(K) = \sum_{k=0}^{K-1} \pi_k^\infty$. Further, the steady-state probability π_k^∞ of state k for an infinite capacity M/D/1 queuing system with load ρ is identified in Page 44 of [8] as

$$\begin{aligned} \pi_k^\infty &= (1-\rho) \left[\sum_{i=1}^{k} e^{\rho i} (-1)^{k-i} \frac{(i\rho)^{k-i}}{(k-i)!} \right. \\ &\quad \left. + \sum_{i=1}^{k-1} e^{\rho i} (-1)^{k-i} \frac{(i\rho)^{k-i-1}}{(k-i-1)!} \right], \quad k \geq 2 \end{aligned} \quad (4)$$

with $\pi_0^\infty = 1 - \rho$ and $\pi_1^\infty = (1-\rho)(e^\rho - 1)$. We note that depending on the choice of ρ, the numerical evaluation of the expression of (4) faces stability issues for the values of k larger than 12. In such cases, the asymptotic approximation of Equation $(15.1.4)$ of [17] can be used to identify the steady-state probabilities as

$$\pi_k^\infty \approx C_0 \left[e^{r_0(k-1)} - e^{r_0 k} \right] \quad (5)$$

where r_0 is the unique negative zero of the equation $r = \rho(1 - e^{-r})$ and $C_0 = \frac{1-\rho}{\rho e^{-r_0} - 1}$.

We note that a similar approach can be applied to the case of an M/G/1/K, or G/G/1/K queue.

III. Optimal Fine Tuning of the RED Parameters

For the discussion of this section, we focus on fine tuning of the RED parameters namely q_{min}, q_{max}, p_{max}, and w_q for a given traffic pattern. We work with fixed size packets and assume a deterministic service time of one packet per unit time.

A. The Case of Instantaneous Queue Size

In this section, we establish a foundation for our optimization problem by focusing on the case of instantaneous queue size, i.e., $w_q = 1$. From the discussion of the previous section and given ρ, one can determine the steady-state probabilities π_k of being in state k where $k \in \{1, \cdots, K\}$. Given such probabilities, the probability of loss for an arriving packet at a RED queue is expressed as

$$P_{LOSS} = \sum_{k=q_{min}+1}^{q_{max}} \pi_k \frac{k - q_{min}}{q_{max} - q_{min}} p_{max} + \sum_{k=q_{max}+1}^{K} \pi_k \quad (6)$$

Note that Equation (6) represents a statistical average in which the probability of loss in each state is calculated based on the queue occupancy in comparison with the RED thresholds. In the presence of a FIFO service discipline utilized by M/D/1/K queuing discipline, the statistical queuing delay of a packet arriving at a RED queue is calculated as

$$\begin{aligned} P_{DELAY} &= \sum_{k=0}^{q_{min}} \pi_k (k+1) \\ &+ \sum_{k=q_{min}+1}^{q_{max}} \pi_k (k+1)\left(1 - \frac{k - q_{min}}{q_{max} - q_{min}} p_{max}\right) \end{aligned} \quad (7)$$

Once more, we note that Equation (7) represents a statistical average in which the delay in each state is calculated based on the queue occupancy and the probability of drop in comparison with the RED thresholds. Utilizing Equation (6) and (7), we can now formulate a primal constrained optimization problem that attempts at minimizing the probability of packet loss subject to an upper bound D_{max} on its statistical queuing delay. As a dual problem, we can also attempt at minimizing the statistical queuing delay of a packet subject to an upper bound on its loss probability.

In what follows we focus on the primal problem. The primal optimization problem is formulated as

$$\min_{q_{min}, q_{max}, p_{max}} \quad P_{LOSS} \quad (8)$$

$$\text{Subject To:} \quad P_{DELAY} \le D_{max} \qquad (9)$$
$$0 \le q_{min} < q_{max} \le K \qquad (10)$$
$$0 \le p_{max} \le 1 \qquad (11)$$

Since q_{min}, q_{max}, and p_{max} appear as decision variables of the optimization problem, solving the problem yields their optimal values.

Generally speaking, the problem above is categorized under NonLinear Integer Programming (NLIP) problems. Since the decision variables appear in the boundary of summations as well as within the expressions, solving the problem is not straightforward. In specific, applying standard numerical optimization approaches such as Sequential Quadratic Programming (SQP) method in conjunction with line search algorithms introduce both convergence and complexity issues. In addition, utilizing Dynamic Programming (DP) to jointly solve for the three decision variables yields high space- and time-complexity results. In order to efficiently solve the problem, we describe a two-step iterative solution to the problem formulated above and prove that our solution is guaranteed to converge to a local minimum.

In the first step, we analytically solve for the optimal value p^*_{max} assuming q_{min} and q_{max} are fixed and given.

Step 1: In the first step, we work with fixed thresholds q_{min} and q_{max}. Thus, the only decision variable in solving the optimization problem is p_{max}. While the cost function is minimized for the smallest value of p_{max}, the constraint function (9) enforces a lower bound on the value of p_{max}. The optimal value of p_{max} is then calculated at the boundary point of the constraint function (9) as

$$\begin{aligned} D_{max} &= \sum_{k=0}^{q_{min}} \pi_k (k+1) \\ &+ \sum_{k=q_{min}+1}^{q_{max}} \pi_k (k+1)\left(1 - \frac{k-q_{min}}{q_{max}-q_{min}} p^*_{max}\right) \end{aligned} \qquad (12)$$

The solution to the equation above appears as

$$p^*_{max} = \left[\frac{\sum_{k=0}^{q_{max}} \pi_k (k+1) - D_{max}}{\sum_{k=q_{min}+1}^{q_{max}} \pi_k (k+1)(k-q_{min})} \right] (q_{max} - q_{min}) \qquad (13)$$

Note that the operation associated with deriving the value of p^*_{max} from Equation (13) has a time complexity in the order of $\mathcal{O}(K)$. Further, the value of p^*_{max} satisfies the constraint function (11).

In the second step, we provide a reduced order search strategy in order to identify the values of q^*_{min} and q^*_{max} based on a fixed value of p_{max} given in the first step.

Step 2: In the second step, we work with a fixed value p_{max} obtained by the solution of step 1. Thus, the decision variables in solving the optimization problem are q_{min} and q_{max}. Considering the fact that q_{min} and q_{max} are integer values when working with the instantaneous queue size and paying attention to the constraint function (10), we propose performing an intelligent search algorithm in the 2D space of (q_{min}, q_{max}). Fig. 2 sketches the feasible region containing $\frac{K(K+1)}{2}$ points in the 2D space of (q_{min}, q_{max}). Hence, the search algorithm has to evaluate the values of the cost function (8) and the constraint function (9) with a fixed p_{max} at $\frac{K(K+1)}{2}$ points to identify the optimal values (q^*_{min}, q^*_{max}). Therefore, the time complexity of the search algorithm is in the order of $\mathcal{O}(K^3)$.

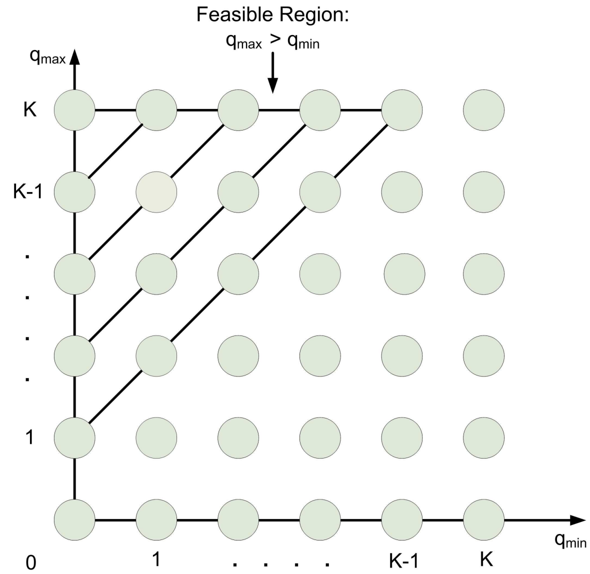


Fig. 2. The feasible region of (q_{min}, q_{max}) in their 2D space.

Considering the staging approach of our solution, one can reach the optimal solution by iteratively applying the result of the second step to the first step and vice versa. Thus, we propose the following algorithm to solve the constraint NLIP problem identified by the cost function (8) and constraint functions (9), (10), and (11).

Iterative Optimization Algorithm

- Step 1: Start from an initial assignment of thresholds (q_{min}, q_{max}) by uniformly splitting the buffer space between them, i.e., $q_{min} = \lfloor K/3 \rfloor$ and $q_{max} = \lfloor 2K/3 \rfloor$. In addition, set the initial iteration number $it = 0$, the maximum number of iterations $it_{max} = 10^6$, and stoppage criterion variables $L_1 = 0$, $L_2 = 1$, $\delta = 10^{-6}$.
- Step 2: Calculate the optimal value of p_{max} from Equation (13).
- Step 3: Identify the optimal values of q^*_{min} and q^*_{max} as follows:
  Reset the intermediate variable L_3 to the value 1.
  $for \quad (q_{max} = 1 \quad to \quad K)$ {
  $for \quad (q_{min} = 0 \quad to \quad q_{max} - 1)$ {
  – If the constraint function (9) is satisfied, calculate the value of the cost function (8) and store it in the intermediate variable L_4.
  – If $L_3 > L_4$, then $q^*_{min} \leftarrow q_{min}$, $q^*_{max} \leftarrow q_{max}$, and $L_3 \leftarrow L_4$.
  } /* $for \quad (q_{min} = 0 \quad to \quad q_{max} - 1)$ */
  } /* $for \quad (q_{max} = 1 \quad to \quad K)$ */
- Step 4: Set $L_1 \leftarrow L_2$ and $L_2 \leftarrow L_3$. If $\frac{|L_1 - L_2|}{L_1} < \delta$ or $it > it_{max}$ STOP.
- Step 5: Go back to Step 2.

We note that the time complexity of implementing the above algorithm is $\mathcal{O}(IK^3)$ where I indicates the number of iterations.

Theorem 1: The two-step iterative optimization algorithm given in this section with decision variables $p_{max}, q_{min}, q_{max}$ converges to a local minimum.

The theorem is a special case of Theorem 2 for which a formal proof is provided in Appendix I.

B. The Case of Average Queue Size

In this subsection, we generalize the formulation of the previous section to the case of average queue size.

We open our discussion by indicating that our objective is to first express the current average queue size q_t in terms of the current instantaneous queue size $\tilde{q}_t$. The latter is equivalent to providing the solution to the first-order difference equation expressed by (1) with input $\tilde{q}_t$ and output q_t. Relying on the method of successive calculations and starting from the initial condition q_0, the following pattern is observed.

$$q_t \;=\; (1-w_q)^t q_0 \;+\; \sum_{k=1}^{t} w_q(1-w_q)^{t-k}\tilde{q}_k \tag{14}$$

Since the equation has a unique solution, it is sufficient to verify that Equation (14) satisfies the original equation. Relying on induction, we start from Equation (1) to note that

$$\begin{aligned} q_t &= (1-w_q)q_{t-1} + w_q\tilde{q}_t \qquad (15)\\ &= (1-w_q)((1-w_q)^{t-1}q_0 \\ &\quad + \sum_{k=1}^{t-1} w_q(1-w_q)^{t-k-1}\tilde{q}_k) \;+\; w_q\tilde{q}_t \\ &= (1-w_q)^t q_0 \;+\; \sum_{k=1}^{t-1} w_q(1-w_q)^{t-k}\tilde{q}_k \;+\; w_q\tilde{q}_t \\ &= (1-w_q)^t q_0 \;+\; \sum_{k=1}^{t} w_q(1-w_q)^{t-k}\tilde{q}_k \end{aligned}$$

arriving at the right hand side of Equation (14).

Analyzing the solution (14), we notice that it consists of a transient and a steady-state term. Considering the fact that $0 \leq 1 - w_q \leq 1$, the transient term $(1-w_q)^t q_0$ goes to zero in steady-state. The steady-state solution is thus expressed as

$$q_t \;=\; \sum_{k=1}^{t} w_q(1-w_q)^{t-k}\tilde{q}_k \tag{16}$$

Our numerical evaluations have supported the observation that the set of discrete random variables $\{\tilde{q}_k\}_{k=1}^t$ are Independently and Identically Distributed (IID) in the steady-state[1]. Recall that the distribution of discrete random variables $\{\tilde{q}_k\}_{k=1}^t$ can be determined from the traffic and queuing profile. Relying on the IID assumption, the steady-state PDF of the random variable q_t appearing in the form of a weighted sum of t random variables $\{\tilde{q}_k\}_{i=k}^t$ can be numerically calculated as a scaled discrete convolution of a number of PDFs, [16]. Further, the PDF of q_t only depends on a small number of random variables q_t, q_{t-1}, q_{t-2}, and so on considering the fact that scaling factor $1-w_q$ is smaller than one.

Once the PDF of q_t is calculated, we can revert back to the constrained optimization problem with the cost function (8) and constraint set (9), (10), and (11). In the latter case, a new constraint related to the variable w_q is added to the constraint set as

$$0 \leq w_q \leq 1 \tag{17}$$

Note that the impact of working with the average queue size on the optimization problem is that the steady-state probabilities appearing in the cost and constraint functions are now depending on the RED parameter w_q representing a new decision variable of the optimization problem. While one can still utilize the two-step recursive optimization approach to solve the resulting problem, the closed-form expression identified for p^*_{max} in the first step does not hold any longer. Rather, a numerical optimization approach such as Sequential Quadratic Programming (SQP) [3] should be used in conjunction with a line search algorithm such as the one proposed by [18] to calculate the values p^*_{max} and w^*_q in the first step.

SQP relies on the Lagrangian theory to convert an optimization problem in its standard form to one without constraints. We define the Lagrangian function of the original problem of the first step as

$$\begin{aligned} L \;&=\; P_{LOSS} + \lambda_1(P_{DELAY} - D_{max}) \\ &\quad + \lambda_2(p_{max}-1) + \lambda_3(w_q - 1) \end{aligned} \tag{18}$$

where the parameters λ_1, λ_2, and λ_3 are the Lagrange multipliers. Having defined the Lagrangian function of the first step, the necessary conditions for optimality are represented by the Karush-Kuhn-Tucker (KKT) conditions described below.

$$\begin{aligned} &\nabla L(p^*_{max}, w^*_q) = 0 \\ &\lambda^*_1((P^*_{DELAY} - D_{max})) = 0 \\ &\lambda^*_2(p^*_{max} - 1) = 0 \\ &\lambda^*_3(w^*_q - 1) = 0 \end{aligned} \tag{19}$$

where $\nabla L = [\frac{\partial L}{\partial p_{max}}, \frac{\partial L}{\partial w_q}, \frac{\partial L}{\partial \lambda_1}, \frac{\partial L}{\partial \lambda_2}]$ and the coefficients ${\lambda_1}^*$, ${\lambda_2}^*$, ${\lambda_3}^*$ are Lagrange multipliers at the local optimum. Positive multipliers indicate active constraints. We now generalize **Theorem 1** to express:

Theorem 2: The two-step iterative optimization algorithm given in this section with decision variables $p_{max}, w_q, q_{min}, q_{max}$ converges to a local minimum.

A formal proof is given in Appendix I.

Next, we investigate potential implications of utilizing average queue lengths rather than instantaneous queue lengths on the second step of our proposed algorithm. In the latter scenario, we note that the search of the second step has to be

[1]Note that the IID assumption does not lead to the loss of generality as we utilize it to reduce the complexity of numerically calculating the resulting PDF of q_t. The PDF of q_t can be numerically calculated even in the absence of the IID property albeit with a higher complexity.

performed over a continuous $K \times K$ space to identify q^*_{min} and q^*_{max}. In order to perform the search over the feasible region of the $K \times K$ space, a quantized grid covering the triangle with edges at coordinates $(0, 0)$, $(K, 0)$, and (K, K) is formed. Therefore, the complexity of the search is much higher depending on the granularity of the quantization grid. The time complexity of implementing the above algorithm is $I \max(\mathcal{O}(K\lfloor\frac{K}{G}\rfloor^2, N \log N)$ where I, G, and N indicate the number of iterations, the grid size, and the degree of quadratic estimation identified by SQP method, respectively.

With a large grid size and when w_q is fixed, e.g., $w_q = 0.002$, the complexity of the problem is reduced to that instantaneous queue size.

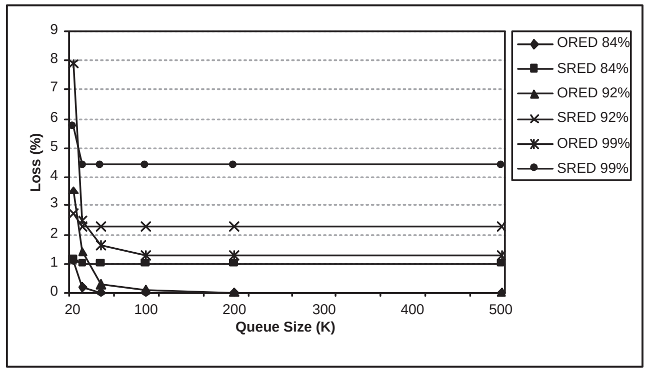
At the end of this section, the following remarks are in order. First, the dual problem of our optimization problem for both cases of instantaneous and average queue size is obtained by swapping the cost function (8) with the constraint function (9). The dual problem can then be solved relying on a similar two-step iterative approach described in this section. Second, it is important to emphasize on the fact the derivation of our optimization results is independent of the traffic profile model as discussed in the Section II-B. In fact, optimization results are valid for as long as the steady-state probabilities of being in each state can be identified either analytically or numerically.

IV. Simulation Results

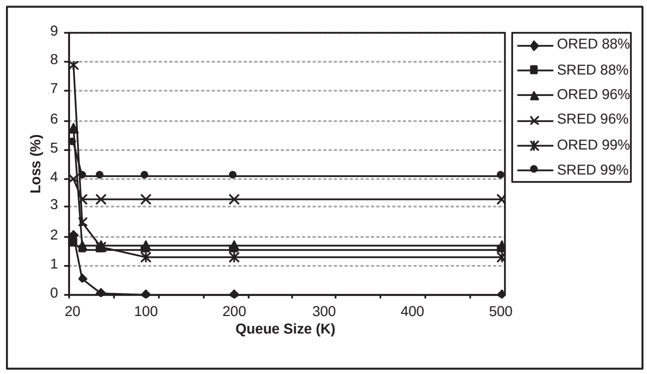
In this section, we validate the performance of our optimal RED algorithm via NS2 [1] simulations. The topology of our experiments includes a single server queue the behavior of which is governed by RED. We experiment with fixed length data packets of size 1024 bytes and in the case of utilizing TCP flows ACKnowledgment (ACK) packets of size 40 bytes. In our experiments, all of the values of q_{min}, q_{max}, and K are *normalized* and expressed as multiples of a size of a data packet. The RED queue is fed with UDP Poisson arrival patterns, FTP arrival patterns utilizing TCP Reno, and HTTP arrival patterns utilizing TCP Reno. The queue is assumed to offer a *normalized* service rate of one packet per second. The latter allows us to examine the performance of our optimal algorithm under the M/D/1/K queuing model as well as TCP traffic patterns mapped to G/G/1/K queuing models.

Viewing the queue capacity K and the maximum delay threshold D_{max} as our design parameters, our experiments span over two sets. In the first set of experiments, we investigate the loss performance of our proposed solution for a fixed D_{max} and varying queue sizes. In the second set of experiments, we investigate the loss performance of our proposed solution for a fixed K and varying delay thresholds. We compare the performance of our solution with that of standard RED. We note that the parameters of standard RED are selected from the default settings of NS2 representing the best heuristic and numerical findings in the literature. Thus, the parameters of standard RED are set at $q_{min} = 5$, $q_{max} = 15$, and $p_{max} = 0.1$. For the case of average queue size, standard RED uses $w_q = 0.002$.

In the discussion and figures below, our optimal RED and standard RED algorithm are referred to as ORED and SRED, respectively. Fig. 3 illustrates the comparison results of ORED with those of SRED for the case of instantaneous and average queue sizes utilizing three different choices of load factor ρ associated with a Poisson arrival pattern. As observed from the figure, the loss performance of ORED is by far better than that of SRED for both cases of instantaneous and average queue size and all three choices of ρ.



(a)

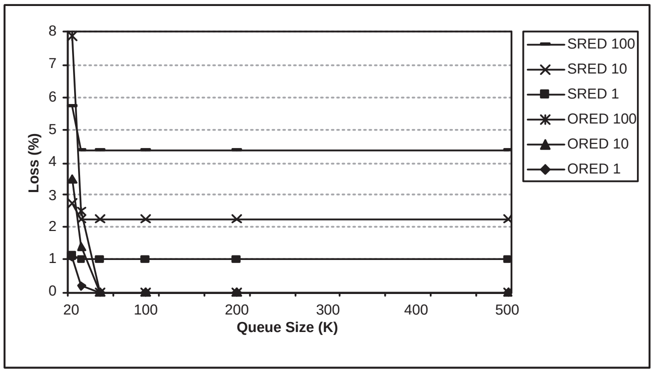


(b)

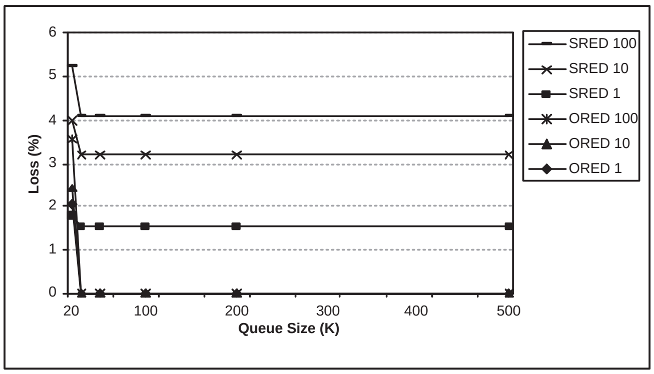
Fig. 3. A performance comparison of ORED and SRED in the case of average queue size, for a fixed normalized service rate of one packet per second, $D_{max} = 100$msec, and a Poisson arrival pattern with load factors $\rho \in \{88\%, 96\%, 99\%\}$. The case of (a) instantaneous queue size with $w_q = 1$, and (b) average queue size are is considered.

Next, we investigate the performance of our scheme in conjunction with TCP traffic where the sources adjust their transmission rate according to the received ACK packets. We note that in the experiments with TCP traffic sources, we numerically generate the PDF of q_t as opposed to utilizing the discussion of Section II-B. Starting with pre-determined values of SRED parameters and adjusting RED parameters as an experiment progresses, our approach works by counting the frequency of queue occupancy as a million packet moving average. Fig. 4 illustrates the results of feeding the queue with FTP and HTTP traffic patterns for the case of average queue size. The three curves in each figure are associated with an aggregate traffic pattern generated by 1, 10, and 100 sources. As observed from the figures, ORED is still by far outperforming SRED.

In the second set of experiments, the queue size is fixed. Utilizing a Poisson arrival with a load factor of $\rho = 0.99$, Fig. 5(a) illustrates the loss performance of ORED as a function of delay threshold for four different values of K. Fig. 5(b) shows similar results for a traffic pattern generated by 10 FTP sources.



(a)



(b)

Fig. 4. A performance comparison of ORED and SRED for aggregate traffic patterns generated by (a) $\{1, 10, 100\}$ FTP, and (b) $\{1, 10, 100\}$ HTTP sources. A fixed normalized service rate of one packet per second, $D_{max} =$ 100msec, and the case of average queue size is considered.

The curves show that increasing the value of D_{max} will result in reducing the loss by servicing more packets staying in the queue for a longer period of time.

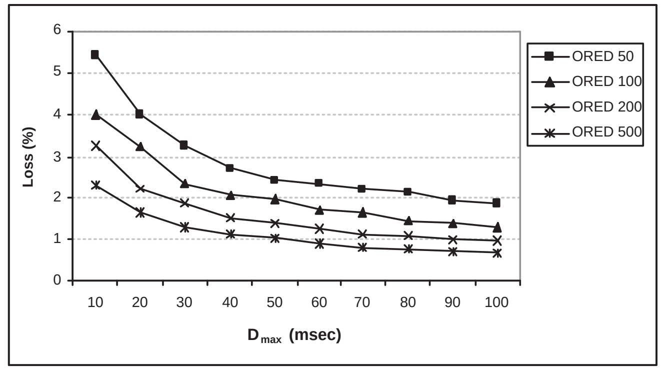
The results of SRED are not shown for clarity but exhibit similar patterns. We have observed that each curve of SRED is always above the curve of ORED for the same choice of K. While not shown here due to space shortage, our experiments in the case of average queue size and in a broad range of parameter selections have led to observing results consistent with those reported here.

Considering the relatively low complexity of our algorithms associated with an average value of $I = 10$ for Poisson experiments and $I = 25$ for TCP experiments[2], we argue that applying our technique to identify optimal settings of RED parameters is highly desired. As the subject of our ongoing work, we are currently in process of investigating the possibility of reducing the complexity of the search algorithm in the second phase of our optimization algorithm.

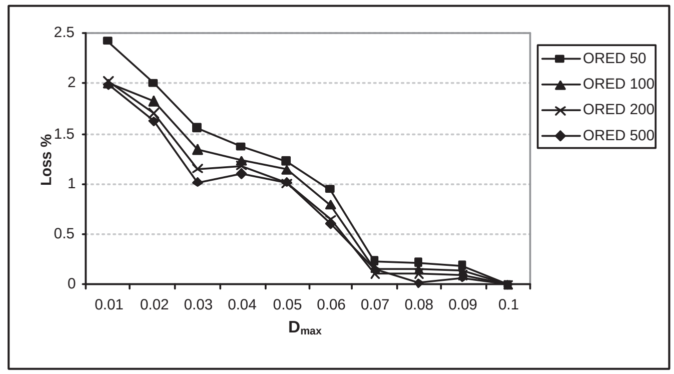
V. Conclusions

In this paper, we presented optimal algorithms for the fine tuning of the parameters of the RED queuing discipline. Our approach spanned over two scenarios working with instantaneous and average queue sizes. For a given traffic pattern and utilizing a statistical analysis approach based on a finite-state Markov chain, we formulated and efficiently solved an optimization problem aimed at addressing the loss and delay tradeoff of a RED queue. Relying on an iterative two-step approach, our solution to the problem identified the optimal settings of the RED parameters. We proved the convergence of our algorithms and investigated their low complexity characteristics. We applied our algorithms to a variety of scenarios, including generic queuing and TCP scenarios, in order to capture their loss and delay performance versus buffer capacity and service rate. Based on our results, we argued that our model is capable of optimally addressing the loss-delay tradeoff of RED queues accommodating time-varying traffic profiles.

[2]We note that the increased number of iterations in the case of utilizing TCP sources is related to applying the moving averaging technique for identifying the PDF of q_t.



(a)



(b)

Fig. 5. A performance comparison of ORED as a function of D_{max} for four different choices of queue size $K \in \{50, 100, 200, 500\}$ with (a) a Poisson arrival pattern identified by $\rho = 0.99$, and (b) an aggregate traffic pattern of 10 FTP sources. The case of instantaneous queue size with $w_q = 1$ is considered.

Appendix I
Proof of Theorem 2

Let us make note of the fact that the cost function of Equation (8) consists of a finite number of terms, one per queue occupancy state. These terms are all positive, with a lower bound of zero. As such, the positive cost function of Equation (8) has a lower bound. Next, we observe that the cost function of Equation (8) can only decrease in each step considering the operating mechanism of the individual phases of our two-step algorithm. Therefore, the sequence of cost function values at each step of the algorithm is a non-increasing sequence with a lower bound. We also note that any non-increasing sequence with a lower bound would converge to a finite number also known as a fixed point. In the case of our optimization problem, converging to a fixed point is equivalent to satisfying the necessary condition for optimality defined below.

The necessary condition for optimality is defined over the threshold values (q^*_{min}, q^*_{max}) and (p^*_{max}, w^*_q) in two steps considering the impact of iterative optimization approach.

First, for a fixed choice of thresholds $(q^{fixed}_{min}, q^{fixed}_{max})$ and the optimal values (p^*_{max}, w^*_q), we have

$$P_{LOSS}(q^{fixed}_{min}, q^{fixed}_{max}, p^*_{max}, w^*_q) \leq P_{LOSS}(q^{fixed}_{min}, q^{fixed}_{max}, p_{max}, w_q) \quad (20)$$

and for every $p_{max} \neq p^*_{max}$ and $w_q \neq w^*_q$. Second, for fixed values $p^{fixed}_{max}, w^{fixed}_q$ and threshold values q^*_{min}, q^*_{max}, we have

$$P_{LOSS}(q^*_{min}, q^*_{max}, p^{fixed}_{max}, w^{fixed}_q) \leq P_{LOSS}(q_{min}, q_{max}, p^{fixed}_{max}, w^{fixed}_q) \quad (21)$$

and for every $q_{min} \neq q^*_{min}$ and $q_{max} \neq q^*_{max}$.

Since, the two-step necessary condition for optimality holds in each individual step of our two-step algorithm, we conclude that it converges to a local minimum. **QED**

References

[1] -, “The Network Simulator - NS2,” Available at http://www.isi.edu/nsnam/ns/.

[2] H.M. Alazemi, A. Mokhtar, M. Azizoglu, “Stochastic Approach for Modeling Random Early Detection Gateways in TCP/IP Networks,” In Proc. IEEE ICC, 2001.

[3] D.P. Bertsekas, “Nonlinear Programming, 2nd Edition,” Athena Scientific Publishing, 1999.

[4] T. Bonald, M. May, J. Bolot, “Analytic Evaluation of RED Performance,” In Proc. IEEE INFOCOM, 2000.

[5] R. Braden, D. Clark, J. Crowcroft, B. Davie, S. Deering, D. Estrin, S. Floyd, V. Jacobson, G. Minshall, C. Petridge, L. Peterson, K. Ramakrishnan, S. Shankar, J. Wroclawski, L. Zhang, “Recommendations on Queue Management and Congestion Avoidance in the Internet,” Internet Draft, March 1997.

[6] V. Firoiu, M. Borden, “A Study of Active Queue Management for Congestion Control,” In Proc. IEEE INFOCOM, 2000.

[7] S. Floyd, V. Jacobson, “Random Early Detection Gateways for Congestion Avoidance,” IEEE/ACM Trans. Networking, August 1993.

[8] E. Gelenbe, G. Pujolle, “Introduction to Queueing Networks,” John Wiley & Sons, Inc., ISBN 0-471-96294-5, 1998.

[9] L. Guan, I.U. Awan, M. E. Woodward, “Stochastic Modeling of Random Early Detection Based Congestion Control Mechanism for Bursty and Correlated Traffic,” In Proc. IEE Software 151, October 2004.

[10] C.V. Hollot, V. Misra, D. Towsley, W. Gong, “Analysis and Design of Controllers for AQM Routers Supporting TCP Flows,” IEEE Trans. Automatic Control, June 2002.

[11] V. Jacobson, “Presentations to the IETF Performance and Congestion Control Working Group,” August 1989.

[12] P. Lassila, J. Virtamo, “Modeling the Dynamics of the RED Algorithm,” In Proc. ACM Quality of Future Internet Services (QoFIS), 2000.

[13] P. Kuusela, P. Lassila, J. Virtamo, P. Key, “Modeling RED with Idealized TCP Sources,” In Proc. IFIP Performance Modeling and Evaluation of ATM & IP Networks, 2001.

[14] R. Laalaoua, T. Czachorski, “Markovian Model of RED Mechanism,” In Proc. CCGRID, 2001.

[15] S.H. Low , D.E. Lapsley, “Optimization Flow Control I: Basic Algorithm and Convergence,” IEEE/ACM Trans. Networking, December 1999.

[16] A. Papoulis, S.U. Pillai, “Probability, Random Variables, and Stochastic Processes,” Fourth Edition,. McGraw-Hill, 2002.

[17] J. Roberto, U. Mocci, J. Virtamo, “Broadband Network Traffic,” Final Report of Action COST 242, Springer Verlag, ISBN 3-540-61815-5, 1996.

[18] D.F. Shanno, “Conditioning of Quasi-Newton Methods for Function Minimization,” Mathematics of Computing, Vol. 24, pp 647-656, 1970.

[19] S. Suthaharan, “Markov Model Based Congestion Control for TCP,” Annual Simulation Symposium, 2004.

[20] Y.C. Wang, J.A. Jiang, R.G. Chu, “Drop Behaviour of Random Early Detection with Discrete-Time Batch Markovian Arrival Process,” In Proc. IEE Software 151, August 2004.

[21] T. Ye, S. Kalyanraman, “Adaptive Tuning of RED Using On-line Simulation,” In Proc. IEEE GLOBECOM 2002.

[22] H. Yousefi’zadeh, H. Jafarkhani, A. Habibi, “Layered Media Multicast Control (LMMC): Rate Allocation and Partitioning,” IEEE Trans. Multimedia, June 2005.